Automatic Formal Analyses of Cryptographic Protocols

19th National Information Systems Security Conference

October 22-25, 1996
Baltimore Convention Center

Dr. Stephen H. Brackin
Arca Systems, Inc.
303 E. Yates St., Ithaca, NY 14850
607-277-8211 or 607-277-2739
brackin@va.arca.com

Supported by ESC/AXS through PRISM, and by Rome Laboratory

Outline of Talk

- **Problem: protocol failure**
- **Automatic Authentication Protocol Analyzer (AAPA)**
- **Three SPX protocols and results of analyzing them**
- **Conclusions, for SPX and arbitrary protocols**

Cryptographic Protocols

- **Goal: Secure communication over insecure networks**
  - **Networks, principals, messages**
  - **Worst case: enemy controls all communication**
  - **Nondisclosure and authentication**
- **Tools:**
  - **Shared or confirmable secrets**
  - **Encryption**
  - **Hash functions**
  - **Timestamps, nonces, signatures, key-exchange functions**
- **Distributed algorithms**

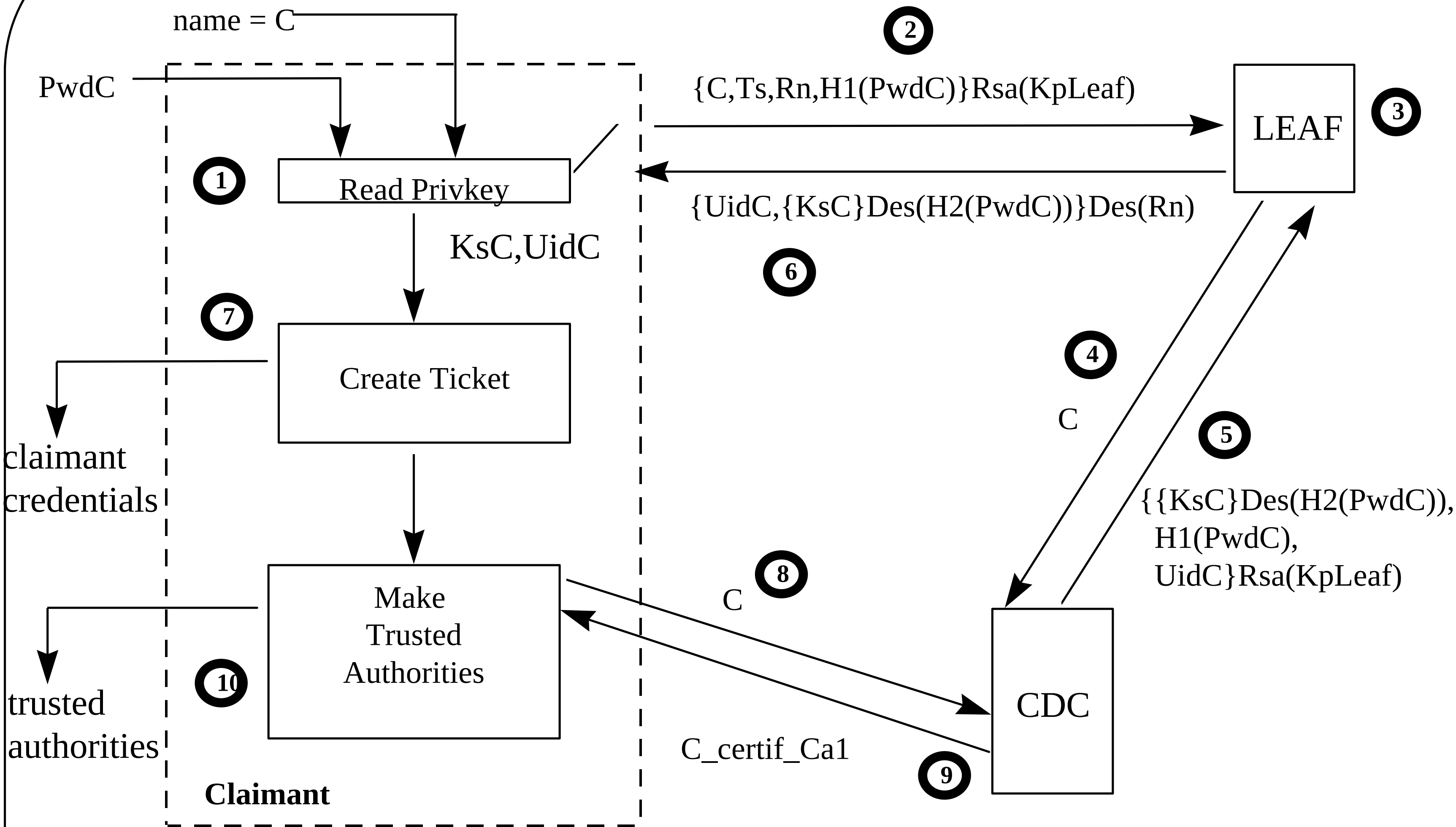
Failure Example

- **Tatebayeshi-Matsuzaki-Newman protocol**
  - **1. A->S: {Ka}Rsa(PkS), A, B**
  - **2. S->B: S,A**
  - **3. B->S: {Kb}Rsa(PkS)**
  - **4. S->A: {Kb}Des(Ka)**
- **AAPA notation, but more-or-less standard**
- **Published (CRYPTO ‘89), recommended by experts**
- **It’s wrong --- and has lots of company**

Automatic Authentication Protocol Analyzer

- **Inputs Interface Specification Language (ISL) specs**
- **Produces Higher Order Logic (HOL) theories**
- **Automatically proves default and user-set goals**
  - **Belief logic extending Gong-Needham-Yahalom logic**
  - **Sample deduction: If P believes only P and Q know K, and P receives M that K decrypts to something meaningful, then P believes Q sent M --- though not necessarily recently or to P**
  - **Proceeds by induction on protocol stage**
- **Gives proof results in ISL**

SPX Credentials Initialization

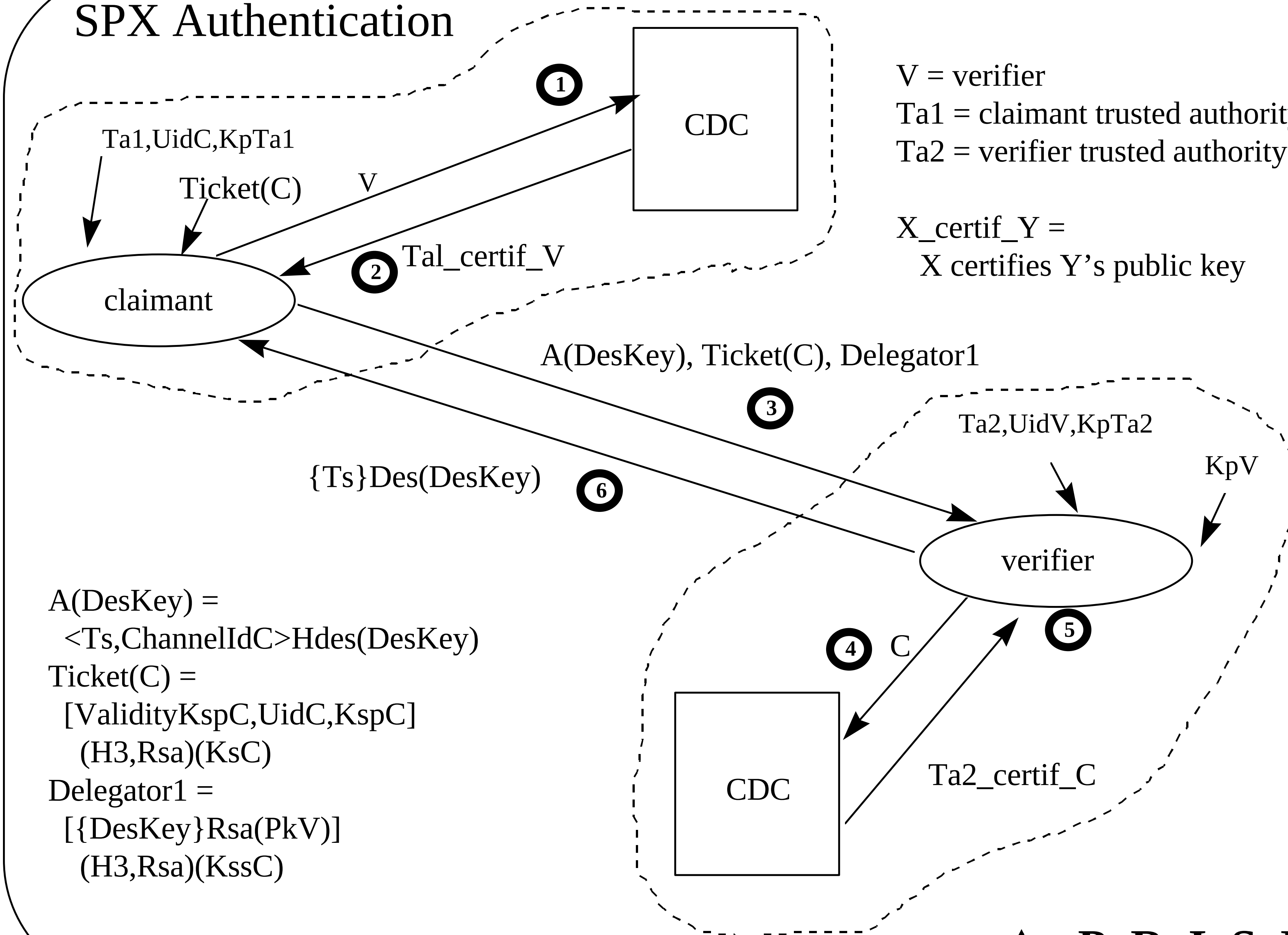


P R I S M
Custom Command and Control

What AAPA Analysis Shows: I

- **KpC must be computable from KsC**
- **Keys must be stored along with recognizable data**
- **PwdC must not be older than KsC**
- **ValidityKpCa1 must include the current time**

SPX Authentication



V = verifier
Ta1 = claimant trusted authority
Ta2 = verifier trusted authority

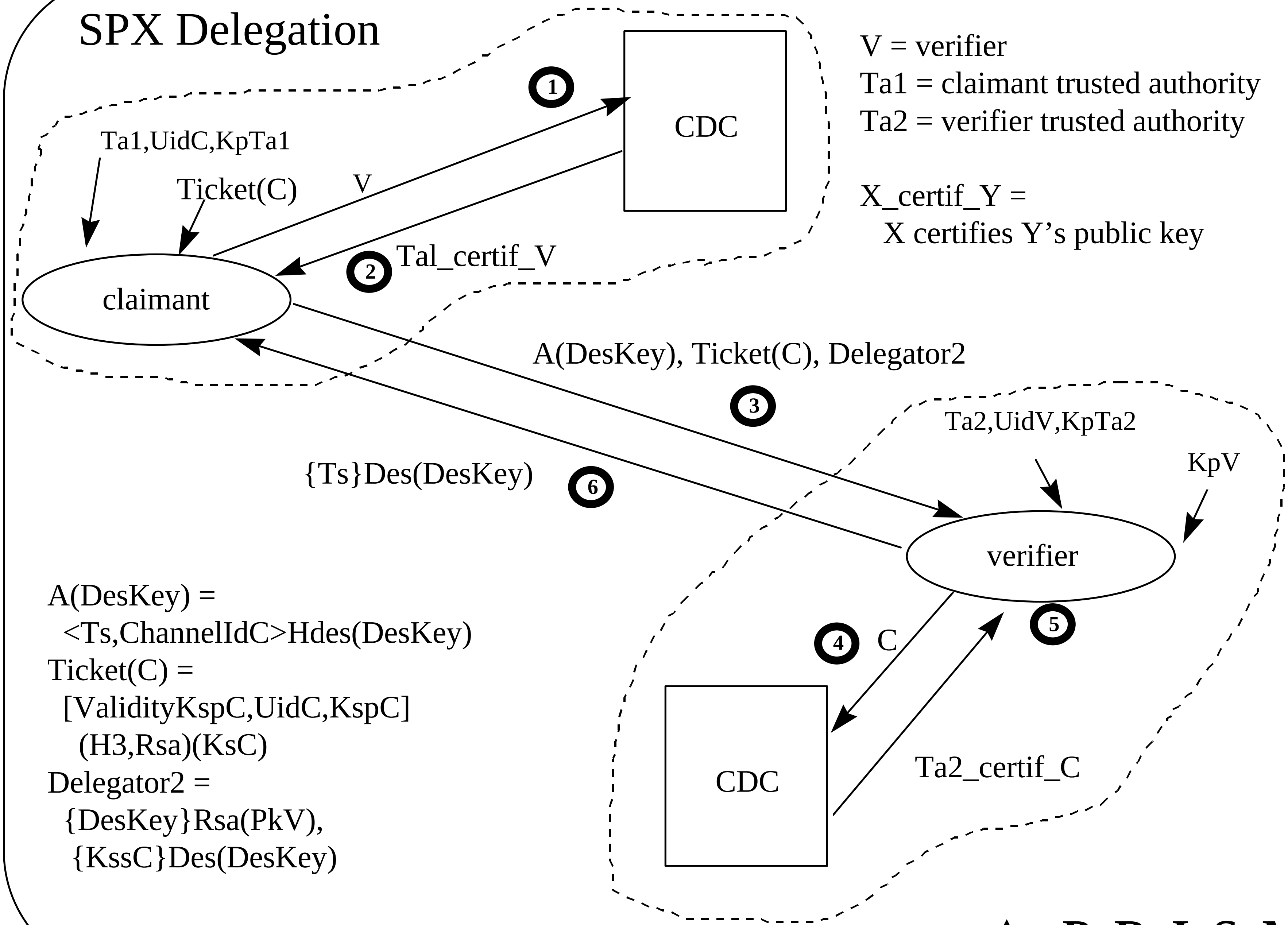
X_certif_Y =
X certifies Y's public key

A(DesKey) =
<Ts,ChannelIdC>Hdes(DesKey)
Ticket(C) =
[ValidityKspC,UidC,KspC]
(H3,Rsa)(KsC)
Delegator1 =
[{DesKey}Rsa(PkV)]
(H3,Rsa)(KssC)

What AAPA Analysis Shows: II

- **Keys must be stored with recognizable data**
- **Validity intervals must include the current time**
  - **ValidityKpV, ValidityKpC, ValidityKspC**
- **Belief DesKey from C depends on dubious assumptions**
- **Delegation gives up to 8 hours of authentication failure**

SPX Delegation



V = verifier
Ta1 = claimant trusted authority
Ta2 = verifier trusted authority

X_certif_Y =
X certifies Y’s public key

A(DesKey) =
<Ts,ChannelIdC>Hdes(DesKey)
Ticket(C) =
[ValidityKspC,UidC,KspC]
(H3,Rsa)(KsC)
Delegator2 =
{DesKey}Rsa(PkV),
{KssC}Des(DesKey)

What AAPA Analysis Shows: III

- **Similar recognizability and interval restrictions**
- **Dubious assumptions don’t give belief KssC from C**
- **Banker can obtain medical records**

Conclusions

- **For the SPX protocols:**
  - **Initialization must include checks for meaningful data**
  - **Authentication possibly flawed**
  - **Delegation possibly flawed**
  - **These issues should be addressed in documentation**
- **For all cryptographic protocols:**
  - **The AAPA is a fast, easy tool for reducing failures**
  - **The AAPA can be used as part of the design process**